jsdf,LLC

Media on the Android Platform

July 12, 2009

Jason Shah
jshah@jsdfllc.com
http://www.jsdfllc.com
Twitter: jasonshah

Agenda

- Who am I? / Goals for today
- What Android gives you
- MediaPlayer
- MediaController
- VideoView
- SoundPool
- References

Who am I?

- Software developer, strategy consultant
- Current: jsdf, LLC: consulting for clients on strategy, technology and development (while working on product ideas…)
- Past:
  - Neuros Technology: headed strategy and new product development
  - Bain & Company: management consulting
  - Trilogy Software: enterprise IT software
- BS, MS in Computer Science UIUC; MBA University of Chicago Booth

Current Android work

- Mediafly Mobile
- CTA Tracker Pro / CTA Tracker Lite
- US National Debt (sample app)

Goals for today

- Introduce the android.media and related classes in the API

- Provide an overview of how to use the basic classes

- Demonstrate android.media with Mediafly Mobile for Android

- Introduce additional resources to get help

Agenda

- Who am I? / Goals for today
- What Android gives you
- MediaPlayer
- MediaController
- VideoView
- SoundPool
- References

What Android gives you

Subject	Section	Description
Codec support	General	• Internal video, audio and image codec support
MediaPlayer	android.media	• Control playback of video/audio files and streams
MediaController	android.widget	• Standard controls for a MediaPlayer
VideoView	android.widget	• Displays a video file.
SoundPool	android.media	• Manages and plays audio resources for applications
AsyncPlayer	android.media	• Plays a series of audio URIs, but does all the hard work on another thread
MediaRecorder	android.media	• Record audio and video
FaceDetector	android.media	• Identifies the faces of people in a bitmap object
AudioManager	android.media	• Change various audio system properties (audio route, volume, ringer, etc.)
AudioTrack	android.media	• Stream PCM buffers directly to the audio hardware for playback

Bold = covered today

Audio codec support

Format	Enc/Dec	Details	File types	Core/G1
AAC LC/LTP	N/Y	Mono/Stereo content in any combination of standard bit rates up to 160 kbps and sampling rates from 8 to 48kHz	3GPP (.3gp) and MPEG-4 (.mp4, .m4a). No support for raw AAC (.aac)	Core
HE-AACv1	N/Y			
HE-AACv2	N/Y			
AMR-NB	Y/Y	4.75 to 12.2 kbps sampled @ 8kHz	3GPP (.3gp)	Core
AMR-WB	N/Y	9 rates from 6.60 kbit/s to 23.85 kbit/s sampled @ 16kHz	3GPP (.3gp)	Core
MP3	N/Y	Mono/Stereo 8-320Kbps constant (CBR) or variable bit-rate (VBR)	MP3 (.mp3)	Core
MIDI	N/Y	MIDI Type 0 and 1. DLS Version 1 and 2. XMF and Mobile XMF. Support for ringtone formats RTTTL/RTX, OTA, and iMelody	Type 0 and 1 (.mid, .xmf, .mxmf). Also RTTTL/RTX (.rtttl, .rtx), OTA (.ota), and iMelody (.imy)	Core
Ogg Vorbis	N/Y		Ogg (.ogg)	Core
PCM/WAVE	N/Y	8- and 16-bit linear PCM (rates up to limit of hardware)	WAVE (.wav)	Core
WMA	N/Y	Supports WMA standard L1-L3 with specific kbps ranges	Windows Media Audio (.wma)	G1

Note: Core/G1 indicates whether codec is guaranteed to work across all devices, or specific to G1 device.
Additional codecs may be supported by future devices.
Source: http://developer.android.com/guide/appendix/media-formats.html

Video codec support

Format	Enc/Dec	Details	File types	Core/G1
H.263	Y/Y		3GPP (.3gp) and MPEG-4 (.mp4)	Core
H.264	N/Y	On the G1, this decoder is limited to baseline profile up to 480x320, and 600 kbps average bitrate for the video stream.	3GPP (.3gp) and MPEG-4 (.mp4)	Core
MPEG-4 SP	N/Y		3GPP (.3gp)	Core
WMV	N/Y	Versions 7, 8 and 9. Simple profile only	Windows Media Video (.wmv)	G1

Note: Core/G1 indicates whether codec is guaranteed to work across all devices, or specific to G1 device.
Additional codecs may be supported by future devices.
Source: http://developer.android.com/guide/appendix/media-formats.html

Agenda

- Who am I? / Goals for today
- What Android gives you
- MediaPlayer
- MediaController
- VideoView
- SoundPool
- References

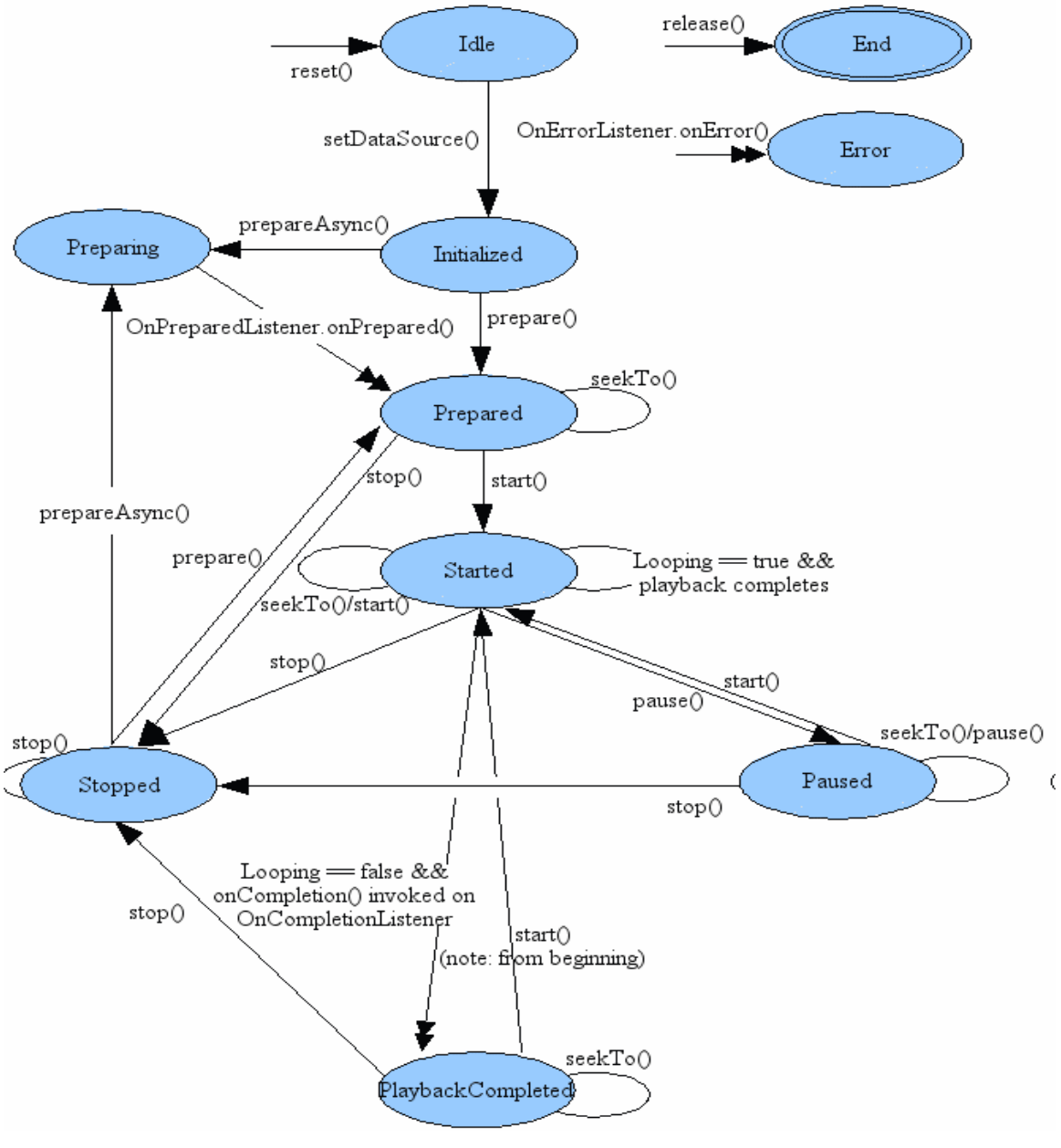
MediaPlayer: Introduction

- MediaPlayer lets you easily control playback of **audio/video files** and **streams**

- Playback source can be:
  - Raw resources bundled with the application
  - Local files
  - http/rtsp streams

- MediaPlayer blocks the UI, so be sure to wrap your own thread around its functions, or use AsyncPlayer

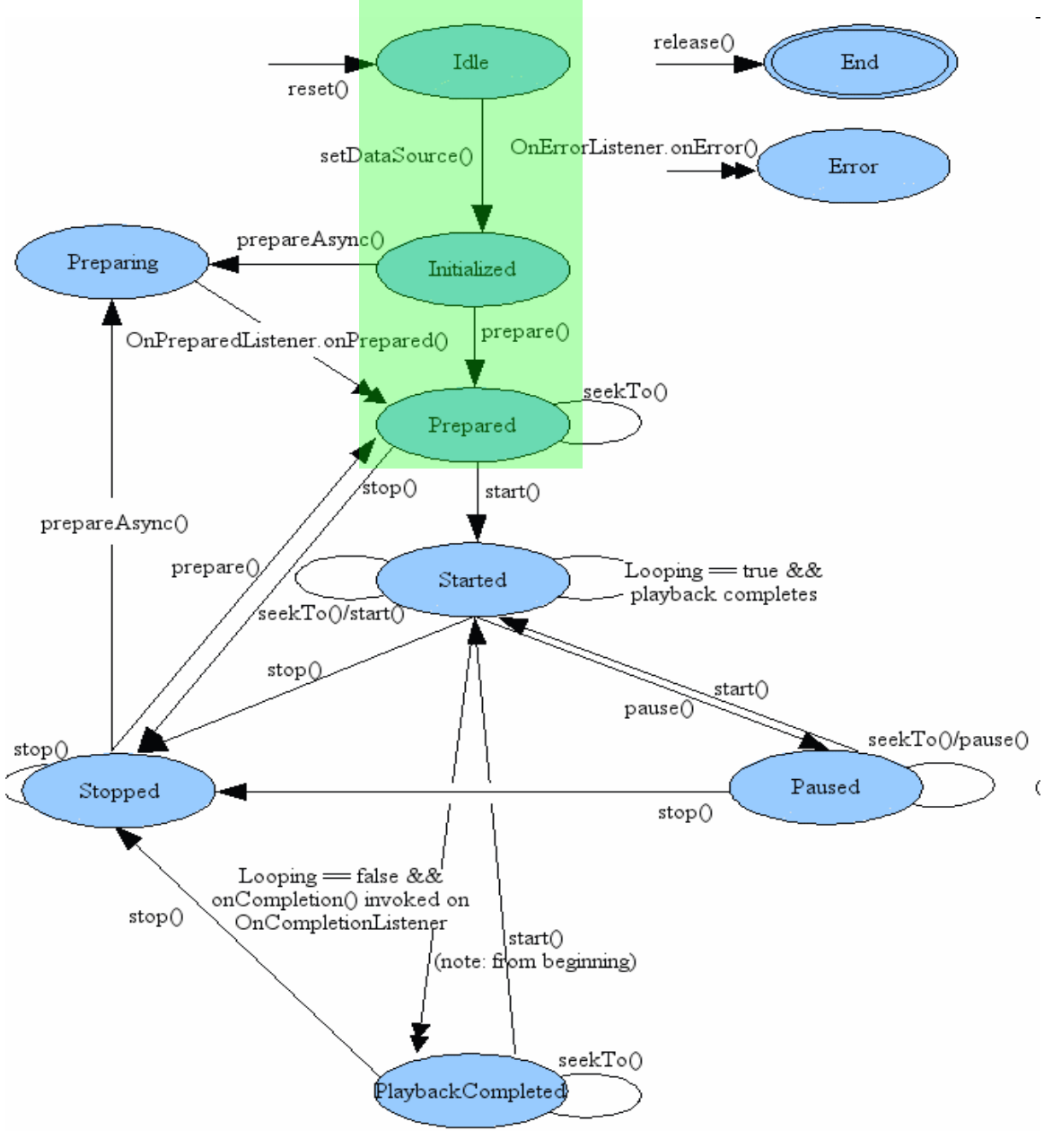
MediaPlayer: State diagram



Notes:

- Single arrow = synchronous
- Double arrow = asynchronous
- Error and End states can be reached from any state in the diagram

MediaPlayer: Create and prepare



Option 1:

```
MediaPlayer mp =
MediaPlayer.create(context,
uri);
```

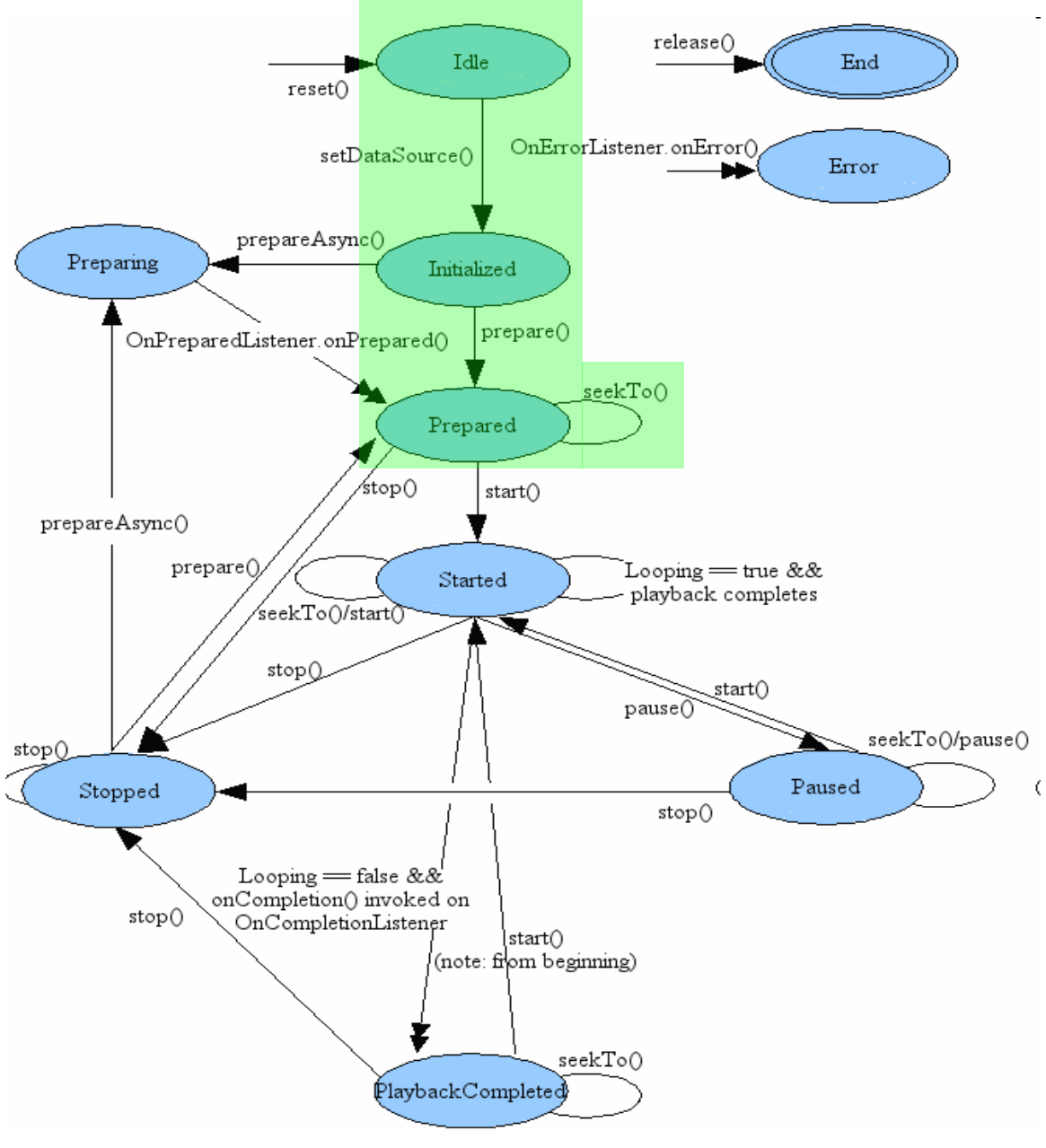
Option 2:

```
MediaPlayer mp;

mp = new MediaPlayer();

mp.setDataSource(uri);

mp.prepare();
```

Notes:

- setDataSource(): sets the data source as URI or file descriptor
- prepare(): Synchronously prepares data for playback by buffering data.
  - Asynchronous version (prepareAsync) also available.
  - Callback OnPreparedListener called if registered on completion

MediaPlayer: Seek to time

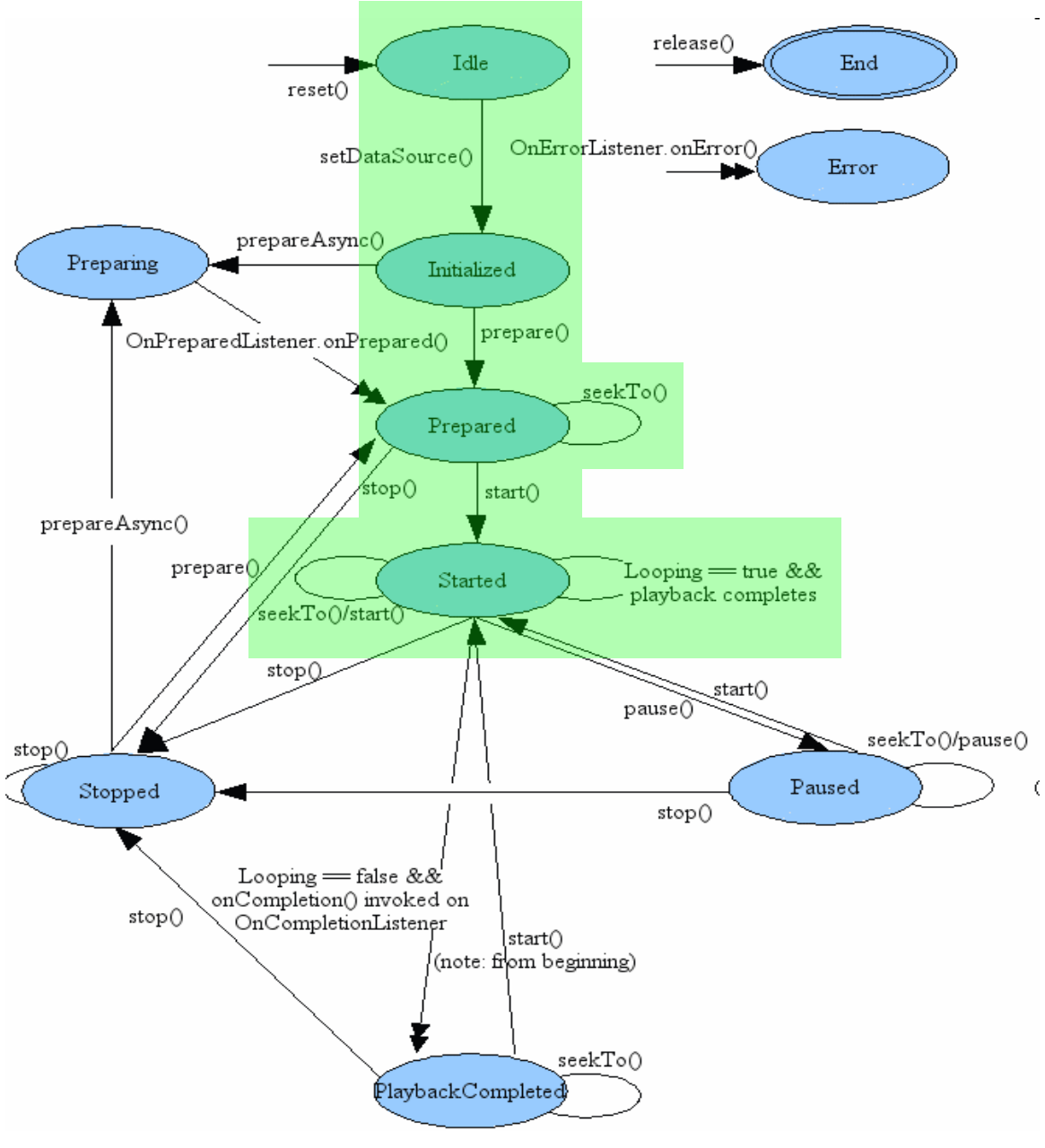



```
mp.seekTo(time);
```

Notes:

- seekTo(): seeks to a specific time in the file in milliseconds.
  - Call returns immediately, but seeking may take several seconds (especially for streams).
  - Callback OnSeekCompleteListener called if registered on completion

MediaPlayer: Start playback




```
mp.start();
```

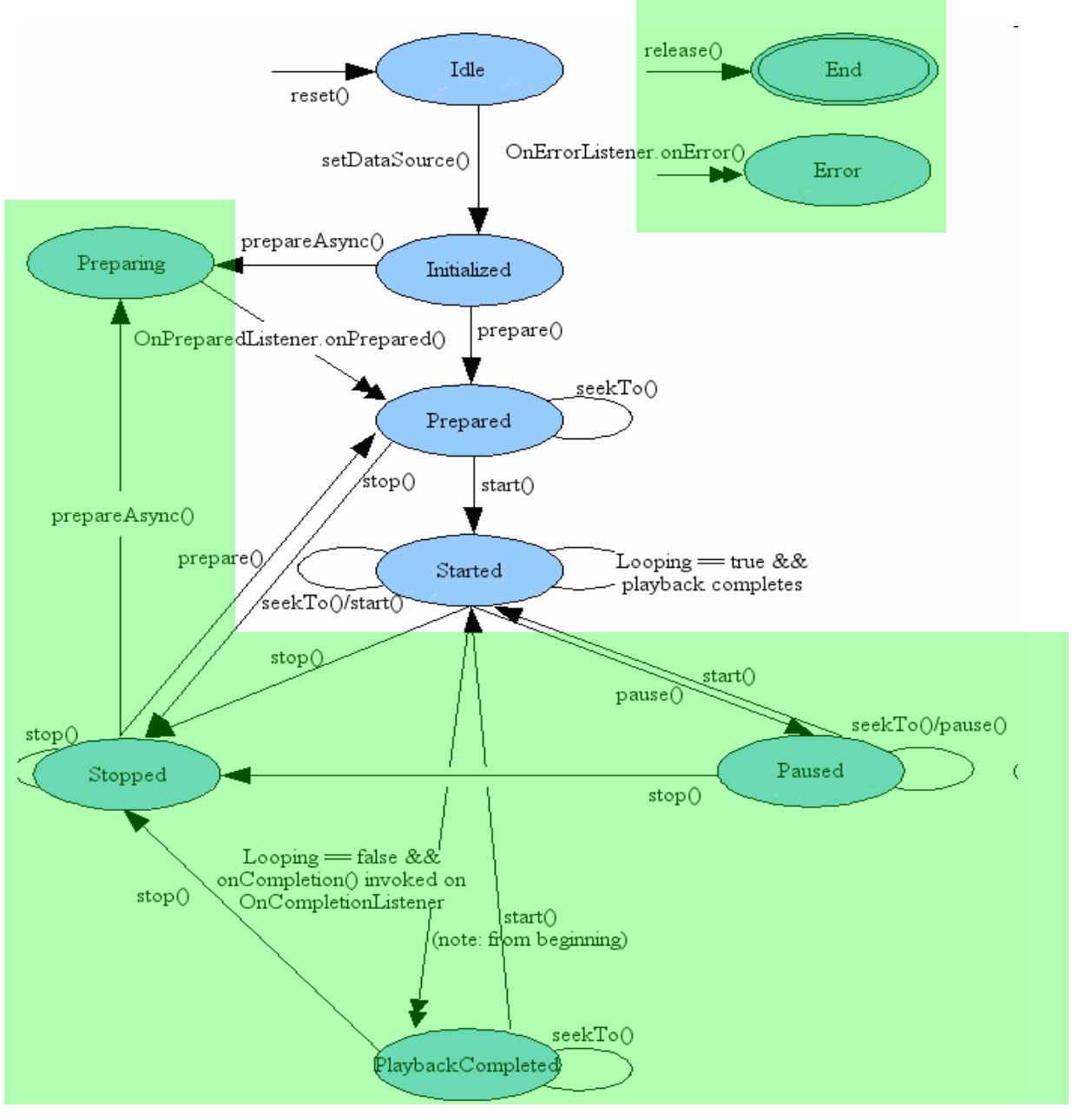
Notes:

- start(): starts playback.

MediaPlayer: demo

- Demo basic playback

MediaPlayer: Additional control



- `mp.pause();`
  - Stops playback and keeps current seek position
- `mp.stop();`
  - Stops playback and loses current seek position
- `mp.seekTo();`
  - Move playhead to new time, buffer new content, and start playing. May take several seconds, especially if streaming.
- `mp.reset();`
  - Move player back to Idle state
- `mp.release();`
  - Move player to End state
  - Release all internal resources held by player

Agenda

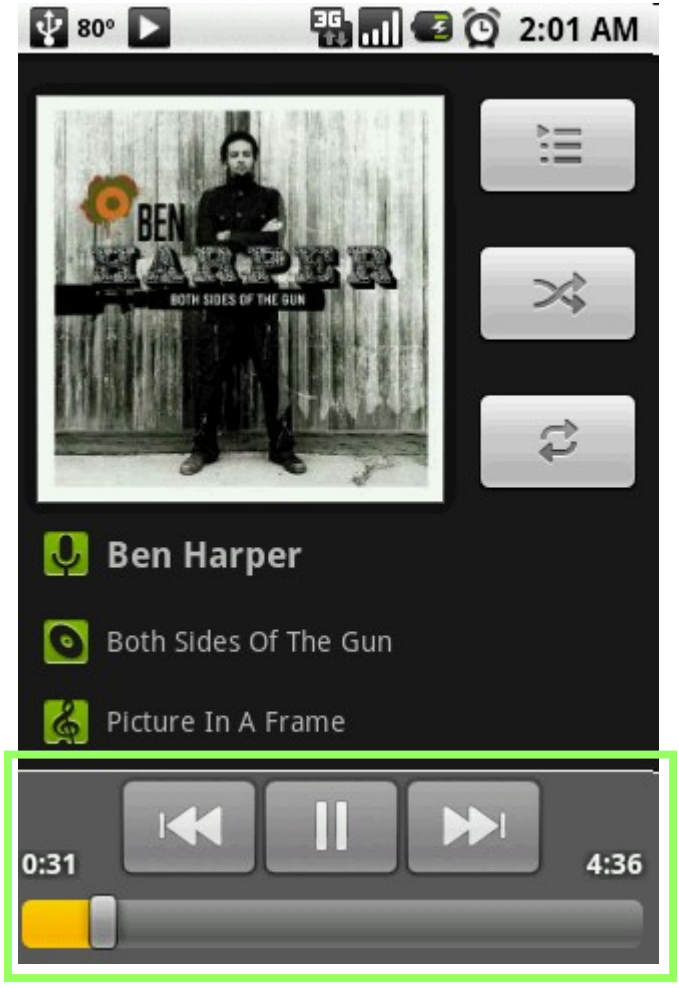
- Who am I? / Goals for today
- What Android gives you
- MediaPlayer
- MediaController
- VideoView
- SoundPool
- References

MediaController: Introduction

- MediaController gives you **standard controls** for a MediaPlayer
  - Play, pause, rewind, fast forward, reverse, forward
  - Seekable progress bar with overlaid playhead and buffer status
  - Handles control updates and synchronization via callbacks

- BUT it is peculiar: it is written primarily to be shown for a period of time then hidden
  - Customization (either by extension or rewrite) may be necessary to achieve your desired behavior

MediaController: Introduction



I'm a MediaController

Image source: http://www.simplehelp.net/2009/07/29/how-to-copy-music-to-your-android-phone-from-windows/

MediaController: Setting it up

- Create and associate your MediaController to an 'anchor' view:

```
MediaController controller = new MediaController(this);

Button showControlsBtn =
(Button)findViewById(R.id.showcontrols);

controller.setAnchorView(showControlsBtn);
```

- When that 'anchor' view is activated, your MediaController will show for 3 seconds

- To show your MediaController manually, call show:

```
controller.show();
```

- MediaController will disappear automatically after 3 seconds by default

MediaController: Handling callbacks

- Register a callback listener to control your MediaPlayer:

```
controller.setMediaPlayer(new MediaController.MediaPlayer() {
    public int getBufferPercentage() {}
    public int getCurrentPosition() {}
    public int getDuration() {}
    public boolean isPlaying() {}
    public void pause() {}
    public void seekTo(int pos) {}
    public void start() {}
});
```

- Connect the callbacks to your MediaPlayer, e.g.:

```
public void start() {
    mp.start();
}
```

MediaController: demo

- Demo MediaController with Mediafly Mobile

Agenda

- Who am I? / Goals for today
- What Android gives you
- MediaPlayer
- MediaController
- VideoView
- SoundPool
- References

VideoView: Introduction

- VideoView attaches to a MediaPlayer to provide default video rendering capabilities for video from various sources
  - Computes its own measurement from the video
  - Can be used in any layout manager
  - Provides scaling, tinting, etc.

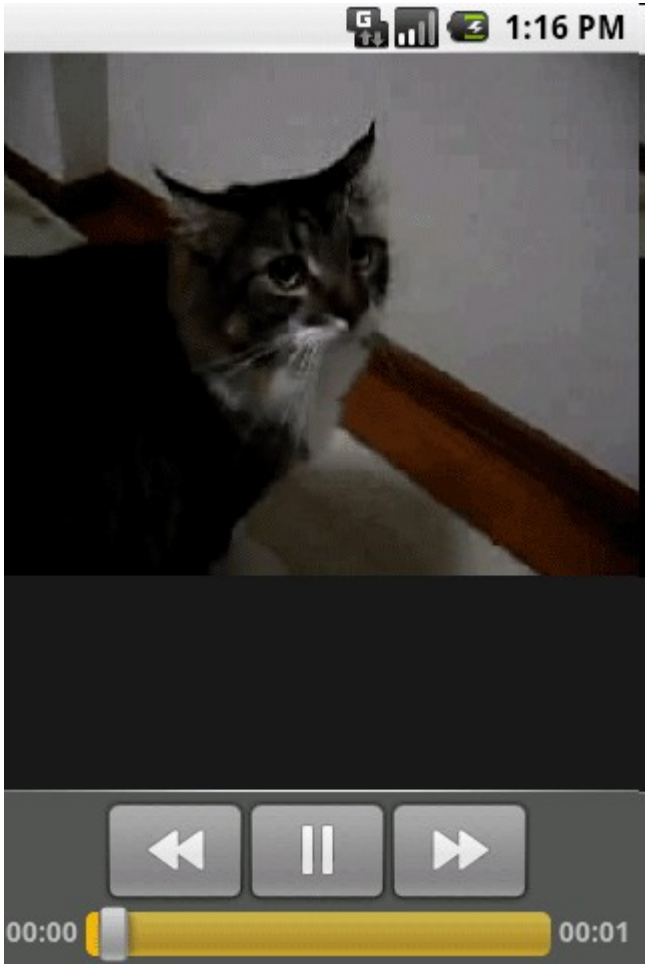


Image source: http://www.saturn.dti.ne.jp/~npaka/android/VideoViewEx/VideoViewEx.gif

VideoView: Setting it up

- Create your VideoView (in XML in this case):

```
<VideoView android:id="@+id/video”
android:layout_width="fill_parent”
android:layout_height="fill_parent” />
```

- Load your video file:

```
File clip=new File("/sdcard/test.mp4");

VideoView video=(VideoView)findViewById(R.id.video);

video.setVideoPath(clip.getAbsolutePath());
```

- Associate your MediaController to VideoView and vice versa

```
MediaController controller = new MediaController(this);

controller.setMediaPlayer(video);

video.setMediaController(controller);
```

- Show your video and its controls

```
video.requestFocus();
```

VideoView: demo

- Demo VideoView

Agenda

- Who am I? / Goals for today
- What Android gives you
- MediaPlayer
- MediaController
- VideoView
- SoundPool
- References

SoundPool: Introduction

- SoundPool manages and plays audio resources for audio-heavy applications/games

- SoundPool:
  - Loads samples into memory from resources or files and decompresses them
  - Manages the number of actively playing streams by assigned priority
  - Allows playback rate and pitch changes real-time

SoundPool: Gaming Use Case

- A typical use case: game has many levels of play, with unique sounds within each level
- The game would:
  - Create a new SoundPool object on level load
  - Load each sound using the appropriate SoundPool.load() object in a thread
  - Perform load early in the level load process to allow time for sound decompression to complete
  - Play sounds with SoundPool.play(); pause, resume, adjust pitch, adjust speed, etc. all as required
  - On level complete, clean up with SoundPool.release()

Agenda

- Who am I? / Goals for today
- What Android gives you
- MediaPlayer
- MediaController
- VideoView
- SoundPool
- References

References

Reference	Source
Android 1.5 API	http://developer.android.com/sdk/1.5_r3/index.html
CommonsWare Busy Coder series, by Mark Murphy: “Android Development” and “Advanced Android Development”	http://commonsware.com/
“Mastering the Android Media Framework” by David Sparks, Technical Lead of Android Media Framework	http://www.youtube.com/watch?v=-0UmSQeWsJc

Questions?